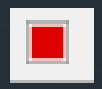
Δρ. Άννα Κεφάλα

WireShark (a network protocol analyser)

WireShark

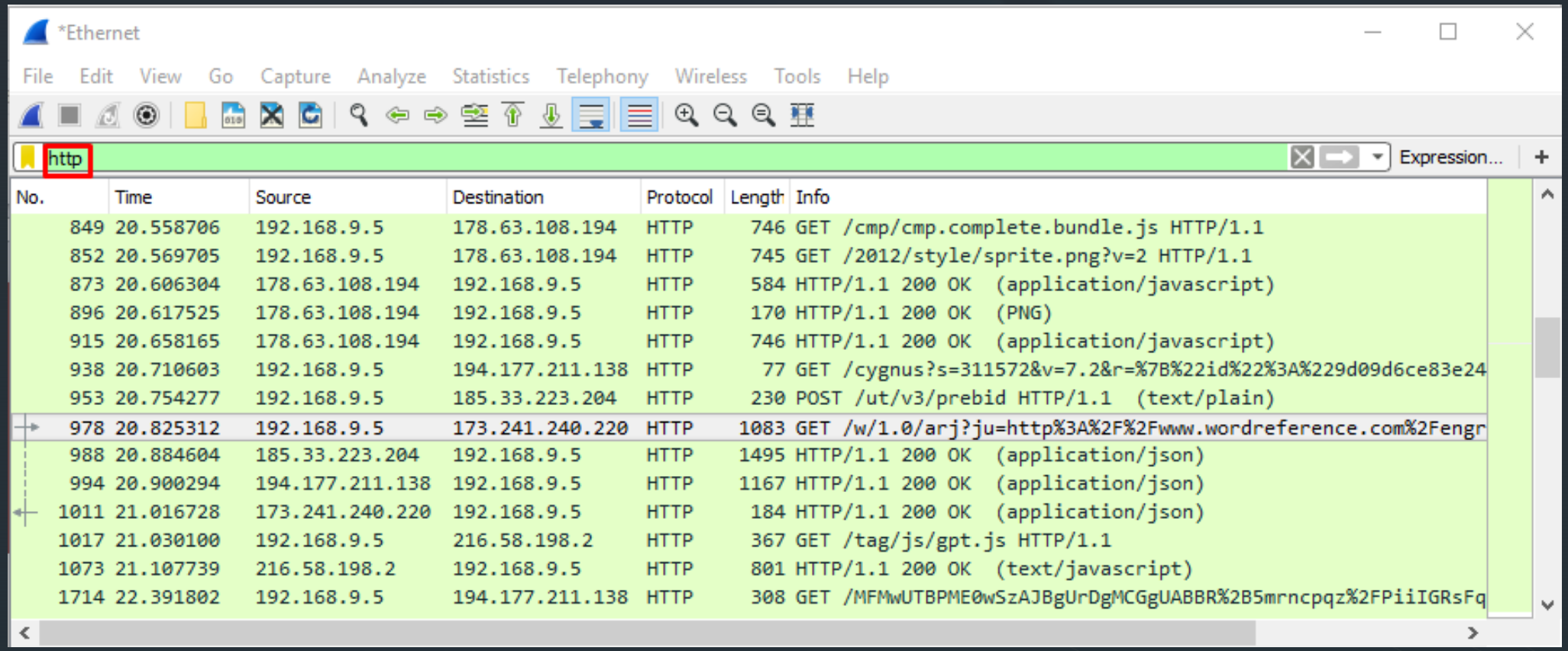
Web Browsing Demo

Try it out!

- Ξεκινήστε τον Browser
- Ξεκινήστε το WireShark
- Επιλέξτε το interface για ανίχνευση
- Ξεκινήστε την ανίχνευση 
- Visit http://ccslab.aueb.gr/ ή κάποιο άλλο URL
- Σταματήστε την ανίχνευση 
- Ανάμεσα στα πακέτα που ανιχνεύθηκαν υπάρχει και η ανταλλαγή των HTTP μηνυμάτων…

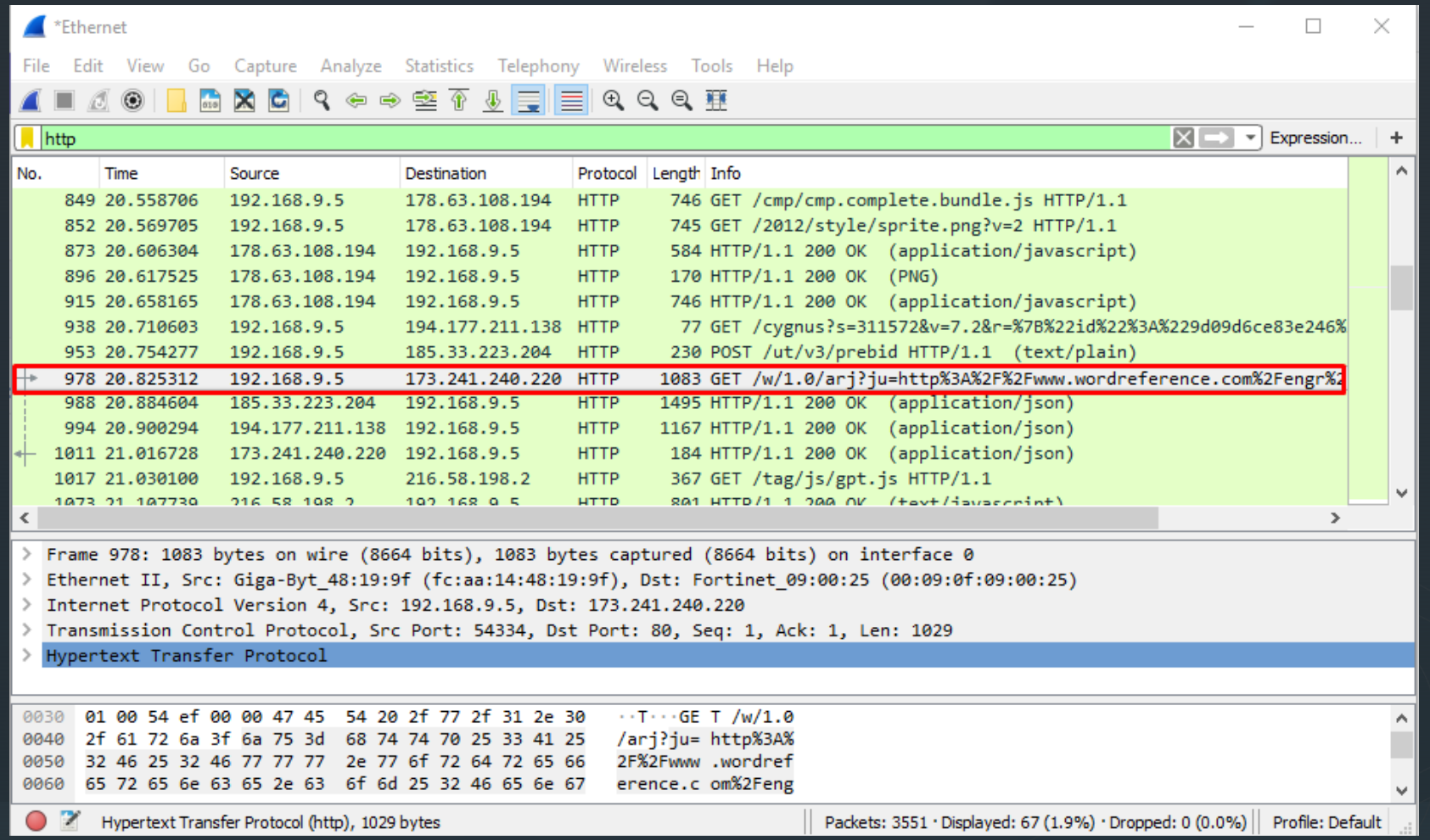
Filtering

Filter HTTP κίνηση



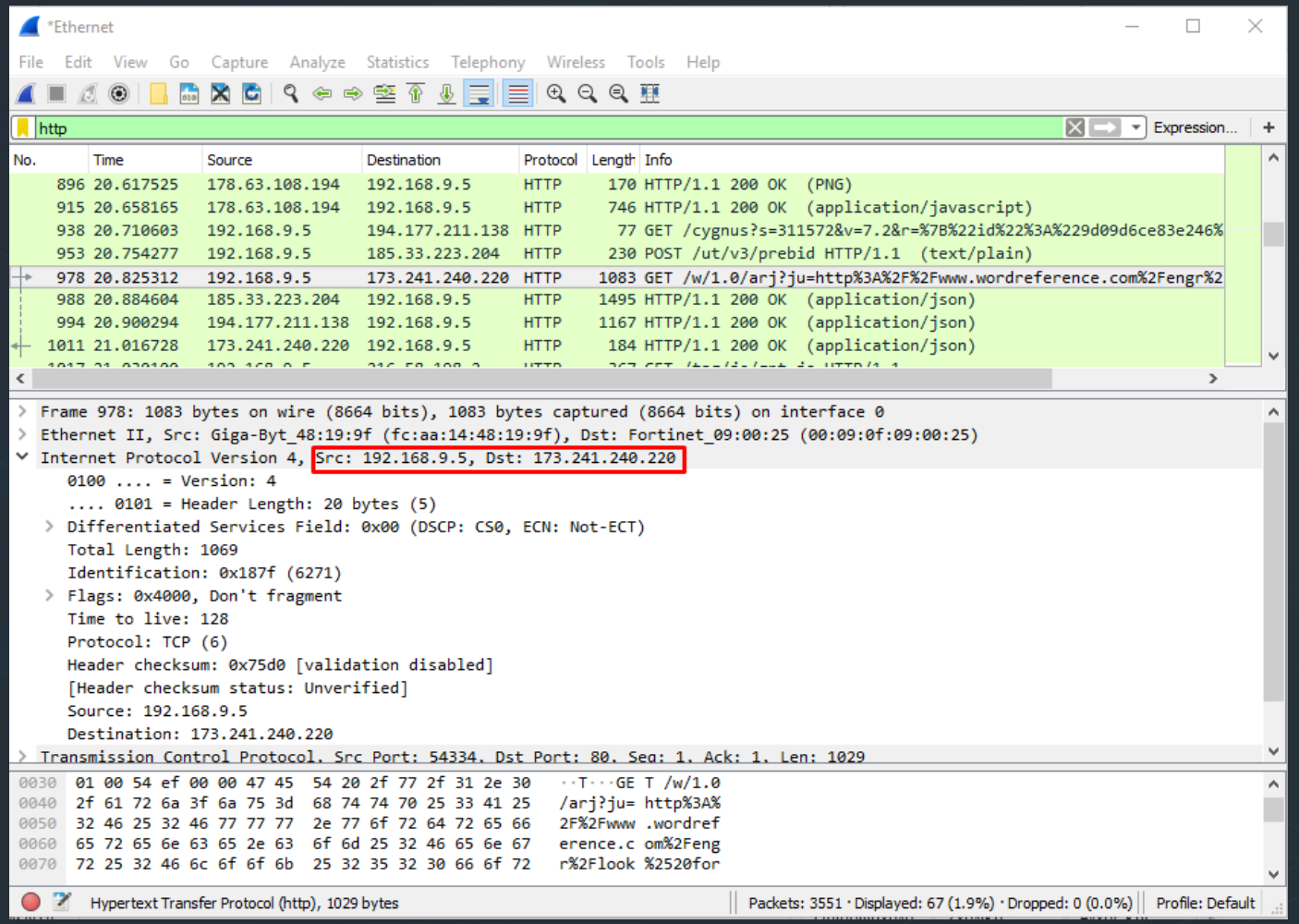
Analyzing

- Επιλέξτε ένα HTTP GET πακέτο



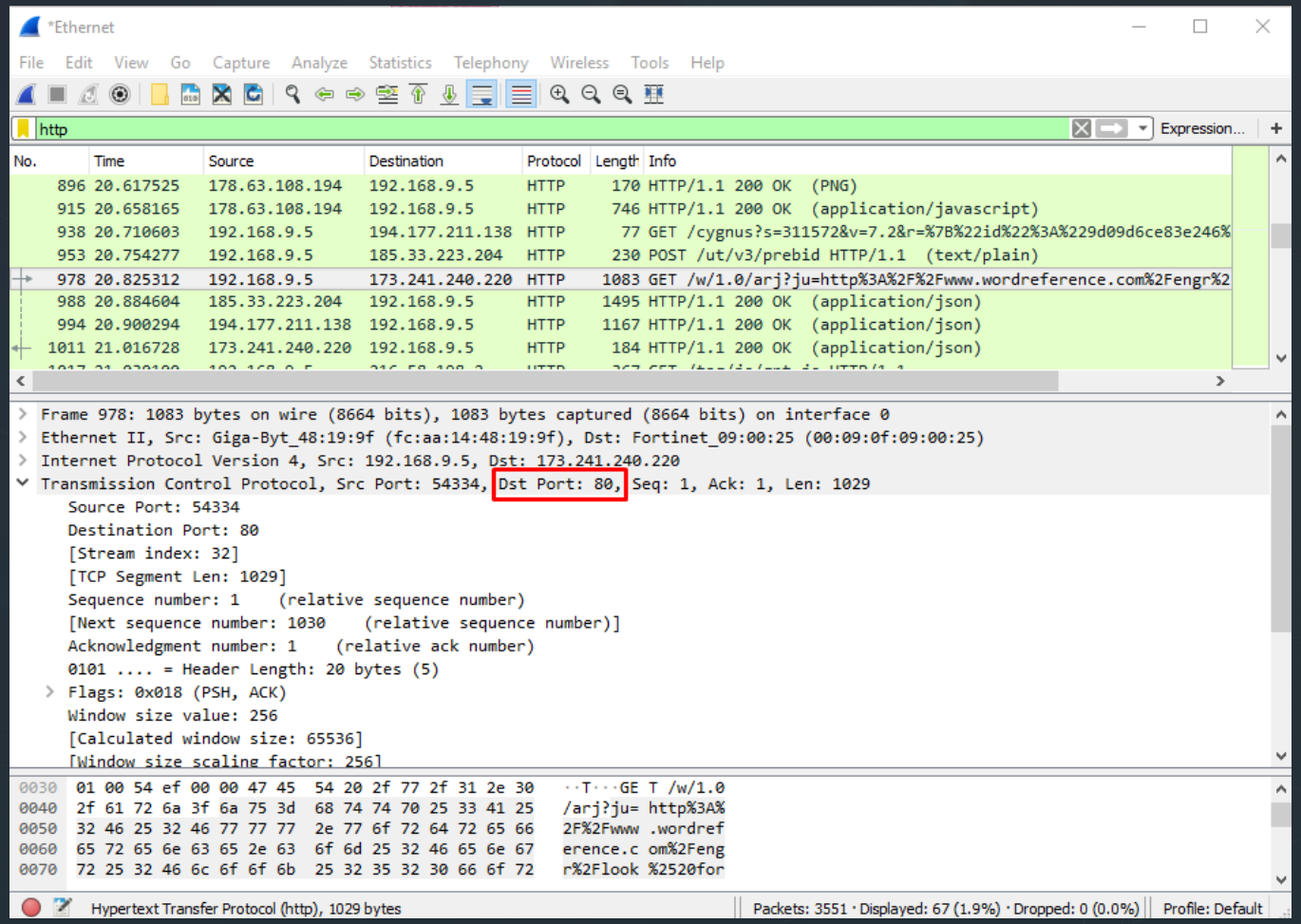
Analyzing IP

- Communicating peers

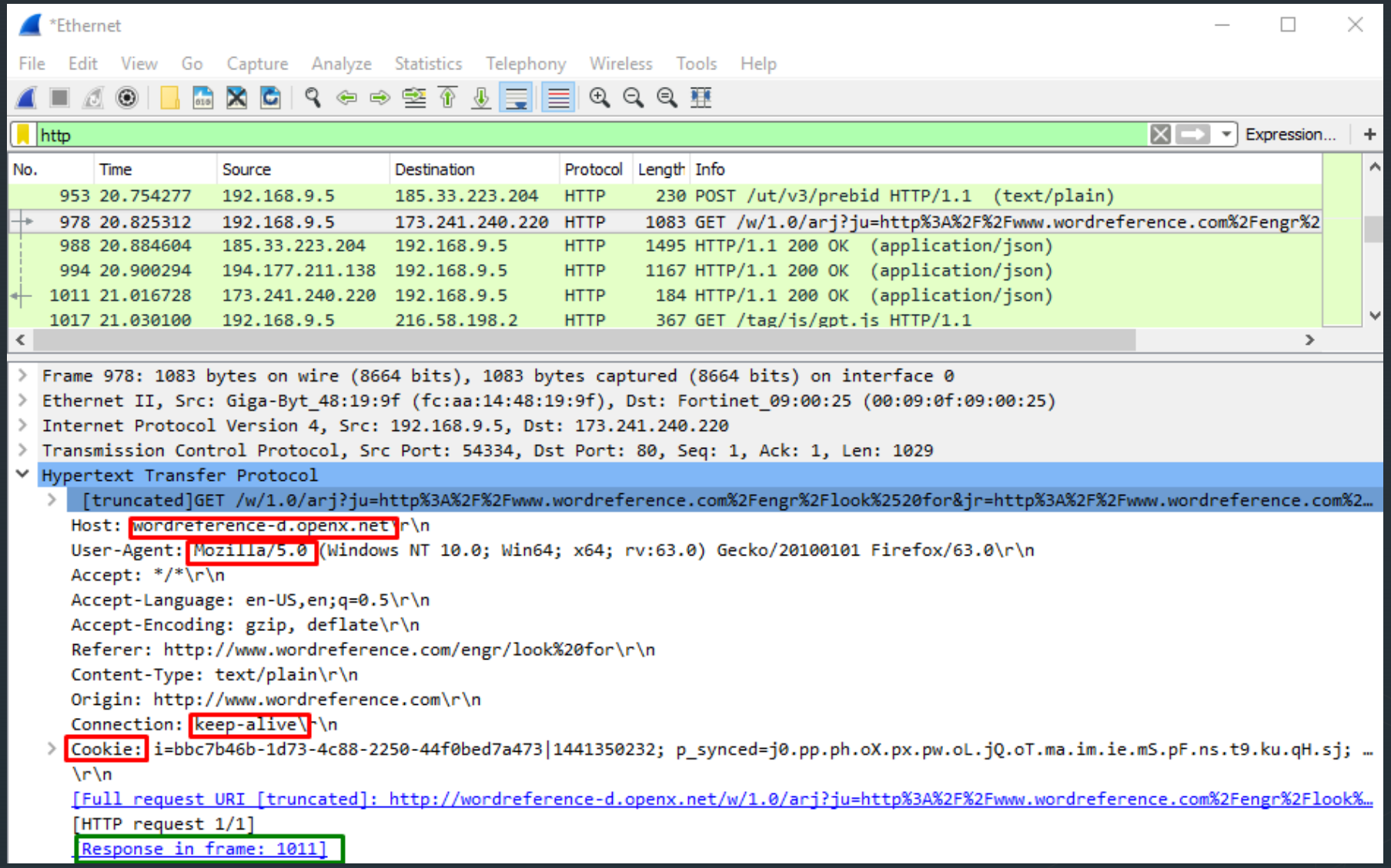


Analyzing Transportation Level

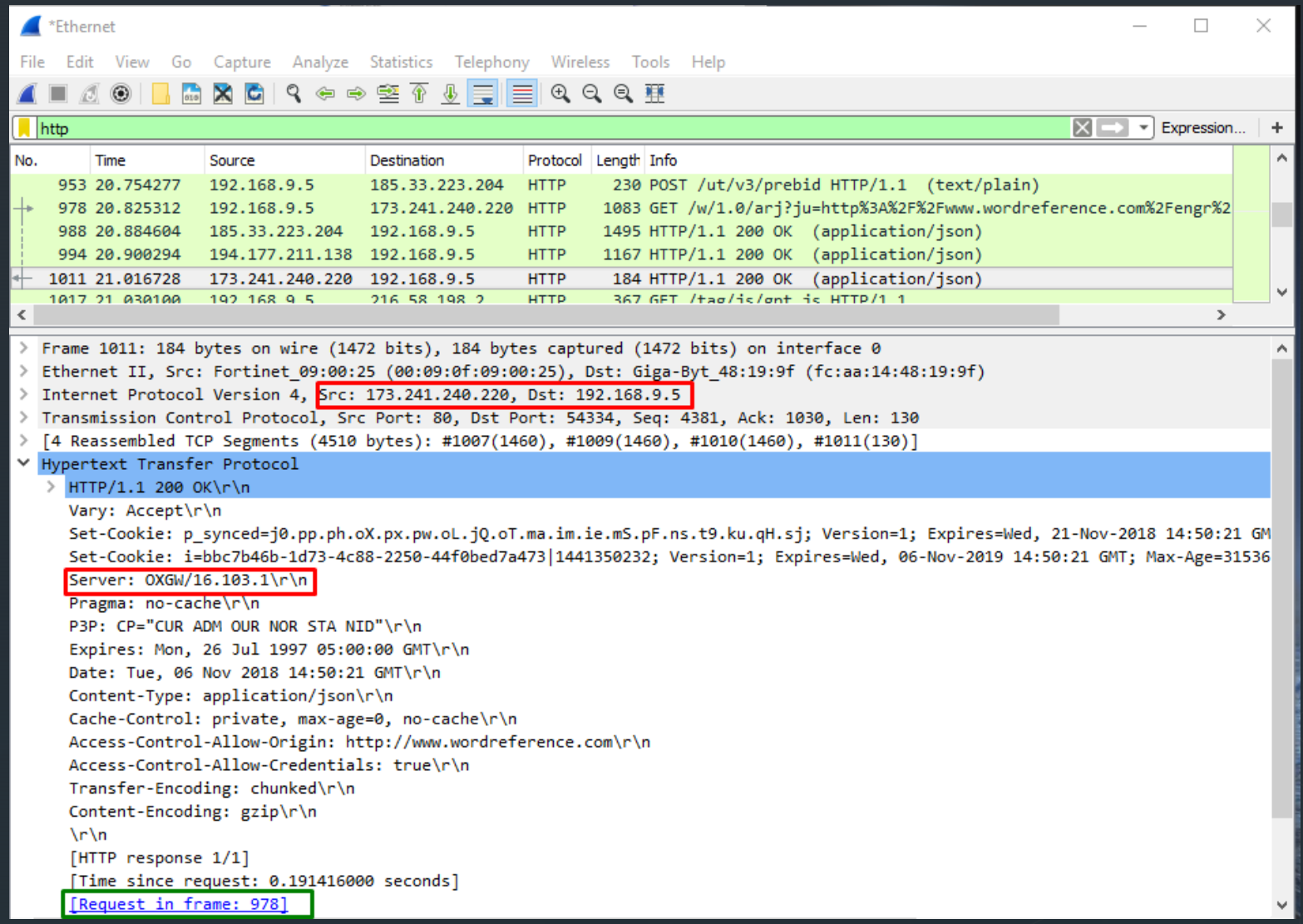
- TCP, well-known port for HTTP?



Analyzing HTTP GET message



Analyzing HTTP response message



WireShark

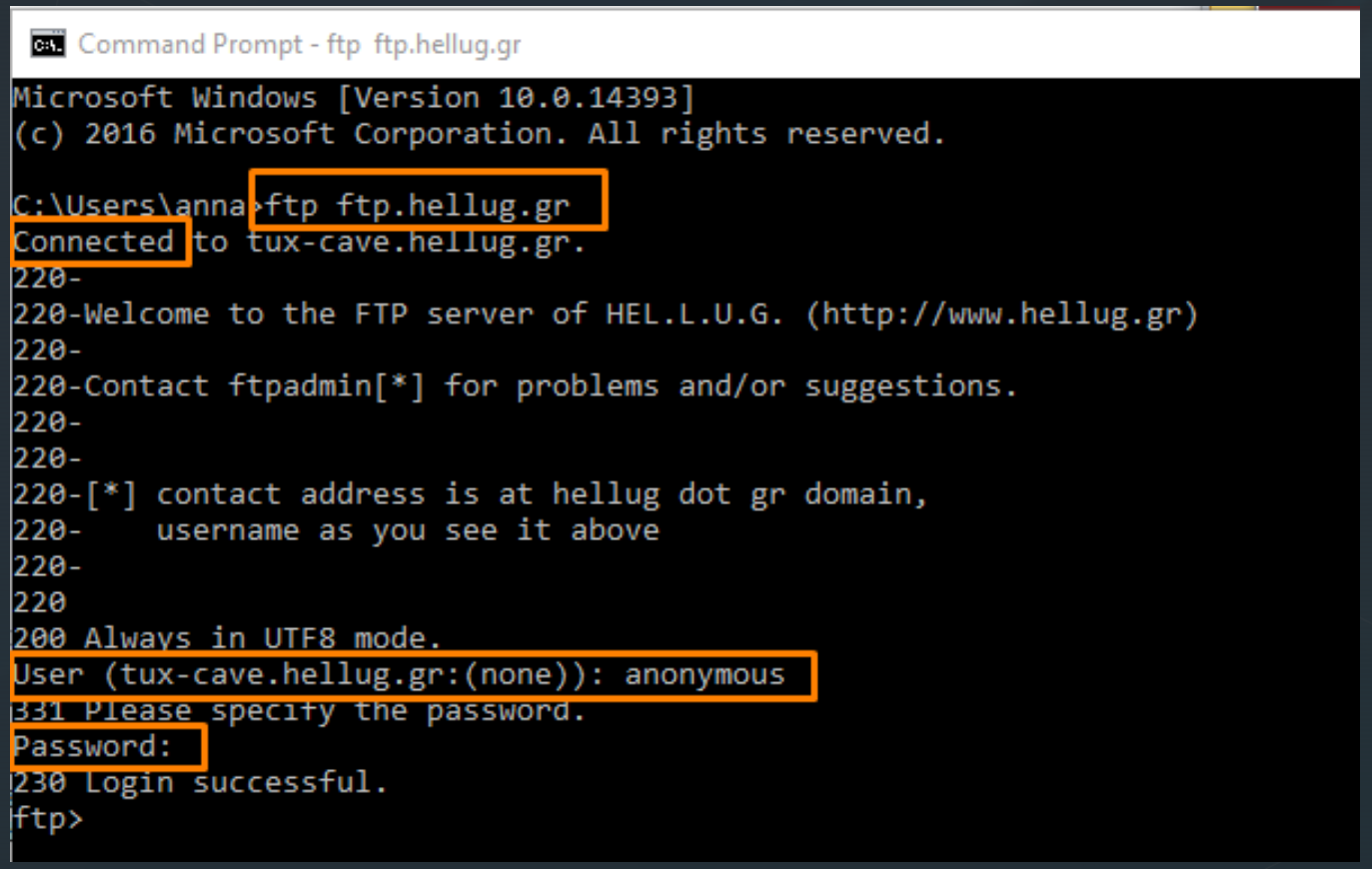
FTP (File Transfer Protocol) Demo

Try it out!

- Ξεκινήστε το WireShark
- Επιλέξτε το interface για ανίχνευση
- Ξεκινήστε την ανίχνευση 
- Ανοίξτε ένα command-line παράθυρο

FTP command-line

- ftp ftp.hellug.gr (user: anonymous, pass: email address)

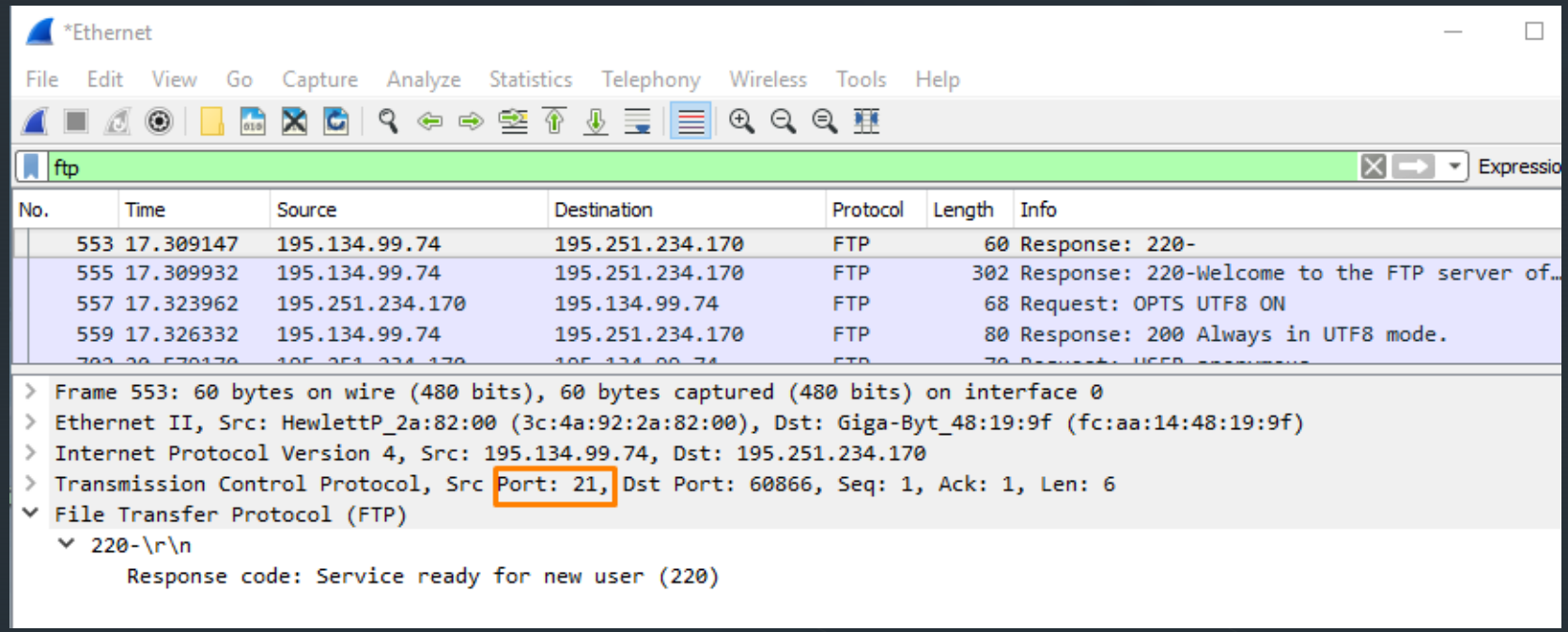


FTP | Get a file

- cd pub/gnutls (change directory)
- get README (file)
- stop capturing

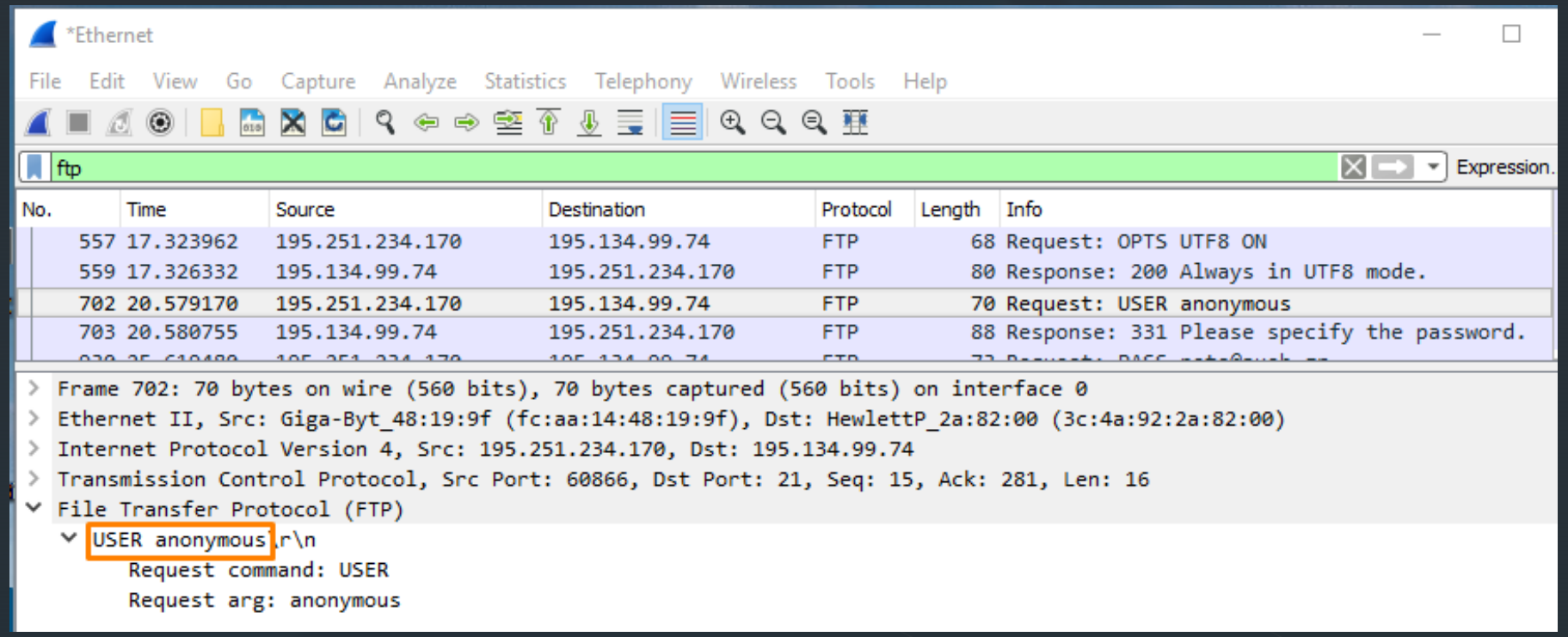
```
ftp>
ftp> cd pub/gnutls
250-More information on GnuTLS can be found at http://www.gnutls.org/
250-
250 Directory successfully changed.
ftp> get README
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for README (67 bytes).
226 Transfer complete.
ftp: 67 bytes received in 0.00Seconds 67000.00Kbytes/sec.
ftp>
```

Filter: ftp

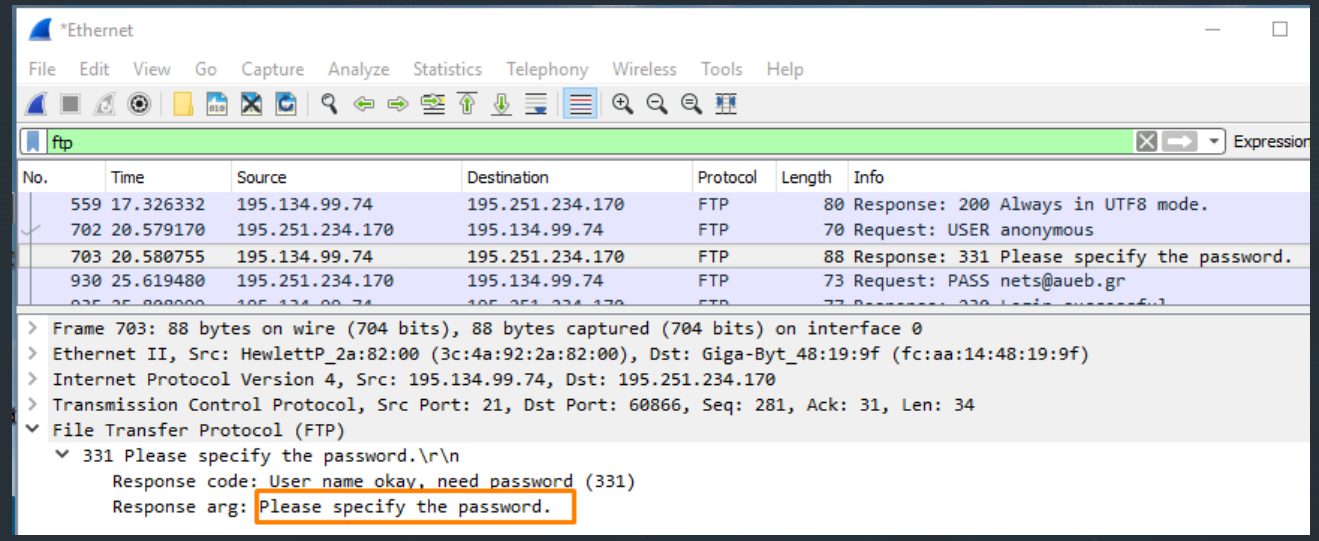


TCP port 21
(well known)

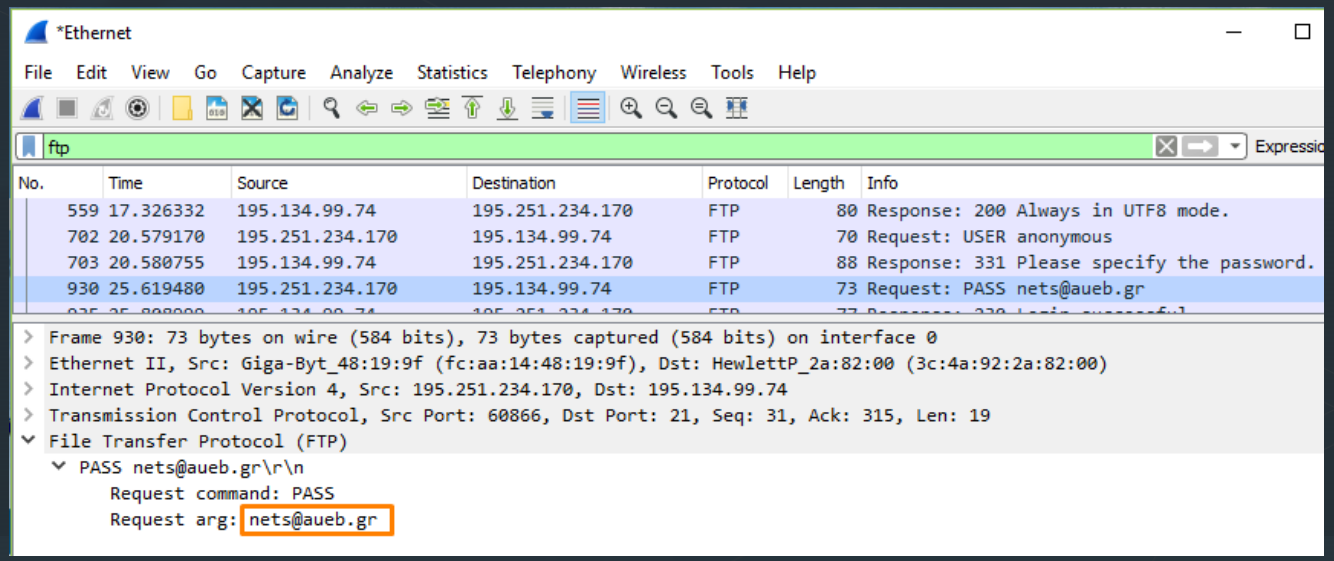
FTP login: *username*



FTP login: *password* (clear text)



Password is intercepted!!!



Retrieve file commands

FTP Protocol

```
File Transfer Protocol (FTP)
  200 PORT command successful. Consider using PASV.\r\n
      Response code: Command okay (200)
      Response arg: PORT command successful. Consider using PASV.
```

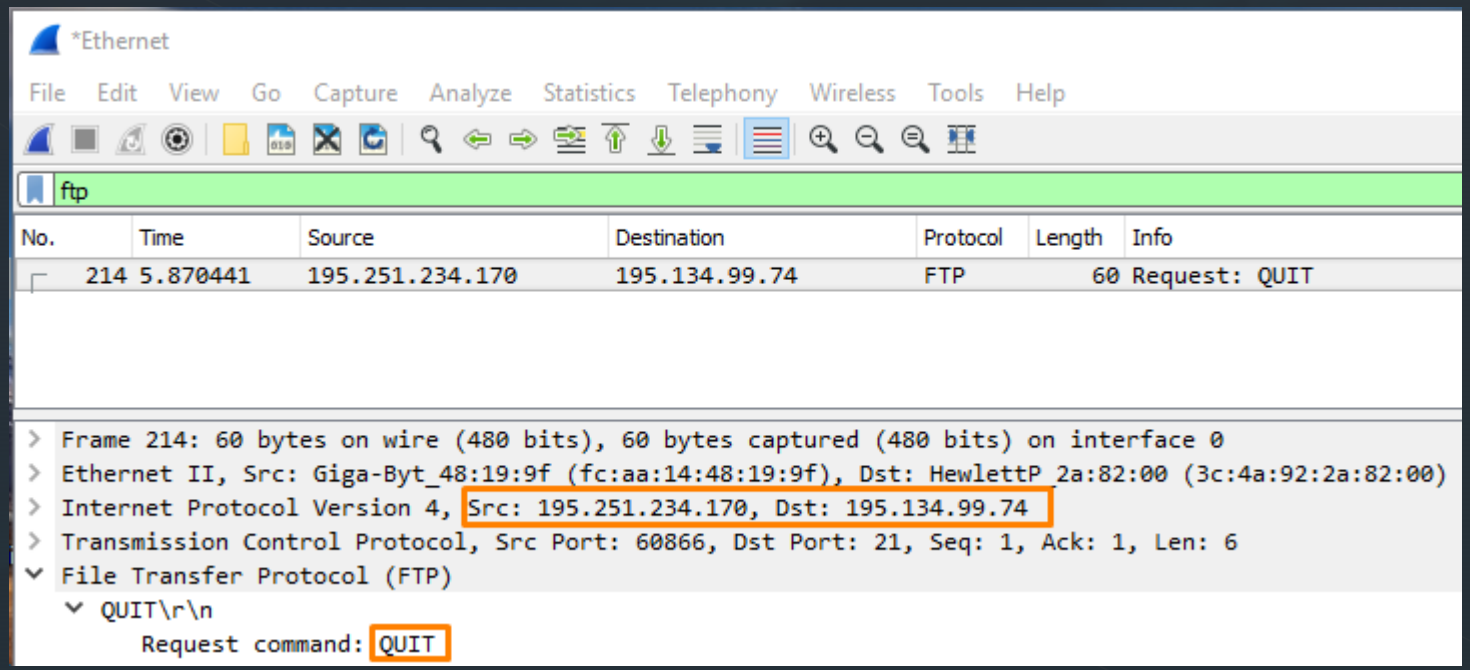
```
Internet Protocol Version 4, Src: 195.251.234.170, Dst: 195.134.99.74
Transmission Control Protocol, Src Port: 60866, Dst Port: 21, Seq: 96, Ack: 503, Len: 13
File Transfer Protocol (FTP)
  RETR README\r\n
      Request command: RETR
      Request arg: README
```

```
Internet Protocol Version 4, Src: 195.134.99.74, Dst: 195.251.234.170
Transmission Control Protocol, Src Port: 21, Dst Port: 60866, Seq: 503, Ack: 109, Len: 64
File Transfer Protocol (FTP)
  150 Opening BINARY mode data connection for README (67 bytes).\r\n
      Response code: File status okay; about to open data connection (150)
      Response arg: Opening BINARY mode data connection for README (67 bytes).
```

```
Internet Protocol Version 4, Src: 195.134.99.74, Dst: 195.251.234.170
Transmission Control Protocol, Src Port: 21, Dst Port: 60866, Seq: 567, Ack: 109, Len: 24
File Transfer Protocol (FTP)
  226 Transfer complete.\r\n
      Response code: Closing data connection (226)
      Response arg: Transfer complete.
```

Quit the session

```
ftp> close
421 Timeout.
ftp> bye
```



Το **traceroute** χρησιμοποιεί το πρωτόκολλο **ICMP** (Internet Control Message Protocol) για να ανακαλύψει τη διαδρομή που ακολουθεί ένα IP πακέτο από τον τοπικό host προς ένα απομακρυσμένο host:
στέλνει μικρά πακέτα με αρχικό TTL=1, και το αυξάνει κατά 1 με κάθε αποστολή πακέτου μέχρι να φτάσει στον τελικό προορισμό. Κάθε φορά που λήγει το TTL, ο κόμβος στον οποίο λήγει, στέλνει πίσω ICMP message (type 11 – TTL-exceeded)
→ μαθαίνουμε την ταυτότητα των ενδιάμεσων δρομολογητών

WireShark

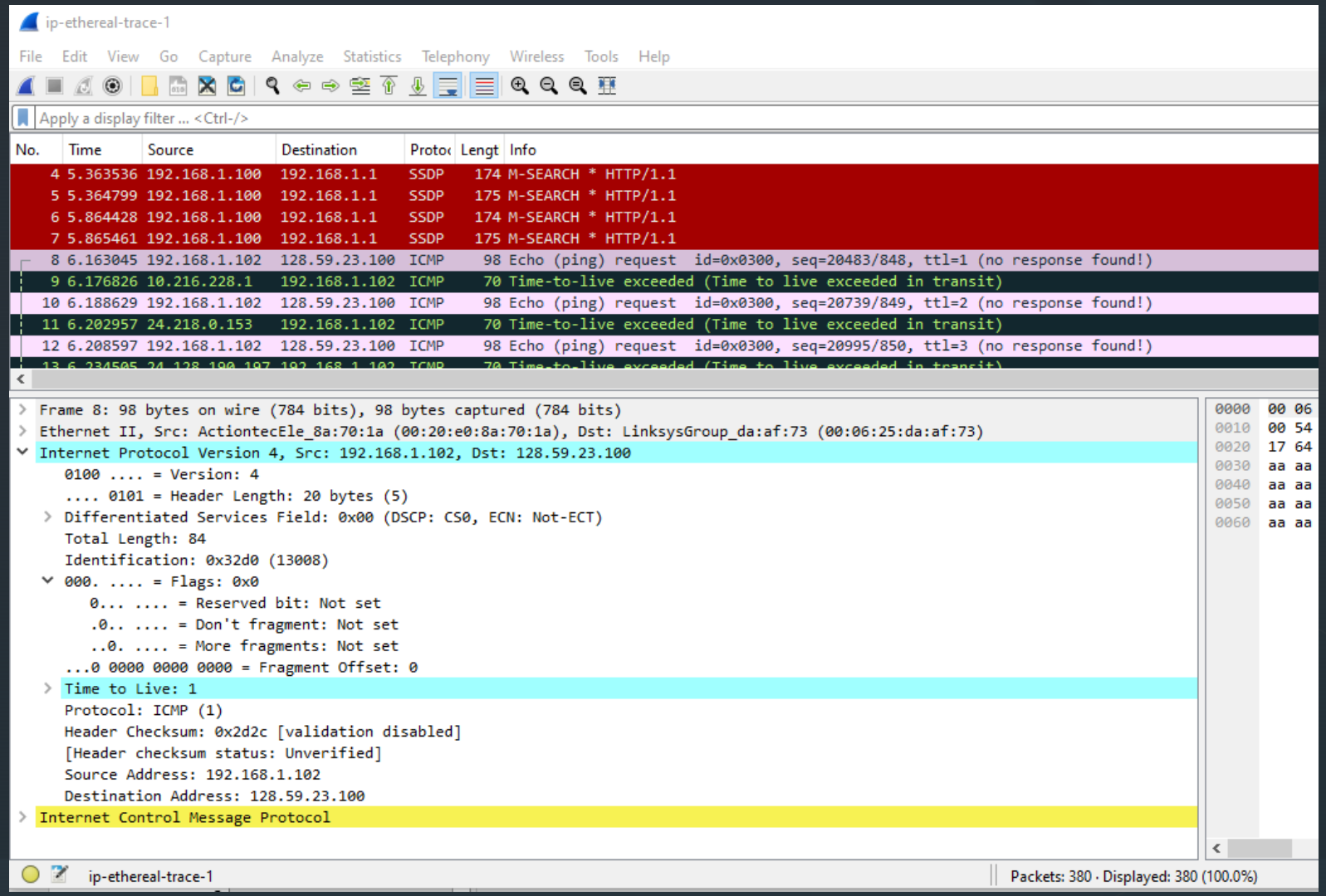
Traceroute Demo

Try it out!

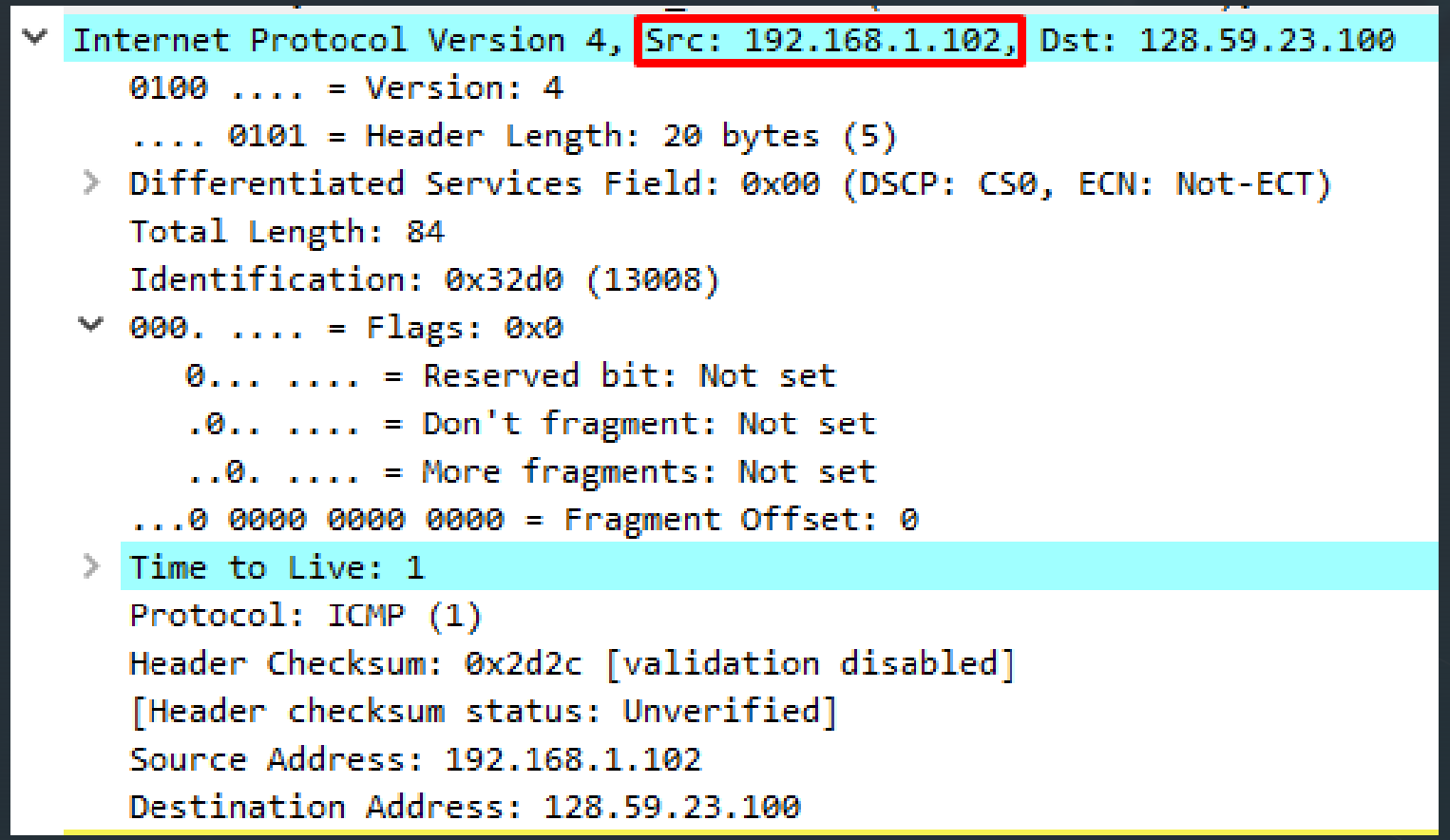
- Ξεκινήστε το WireShark
- Ανοίξτε ένα παράθυρο με command prompt
- Επιλέξτε το interface για ανίχνευση
- Ξεκινήστε την ανίχνευση 
- Στο command prompt παράθυρο δώστε την εντολή:
  `tracert www.acm.org` (*windows*) ή
  `traceroute www.acm.org` (*linux, Mac OS*)
- Σταματήστε την ανίχνευση 

IP protocol

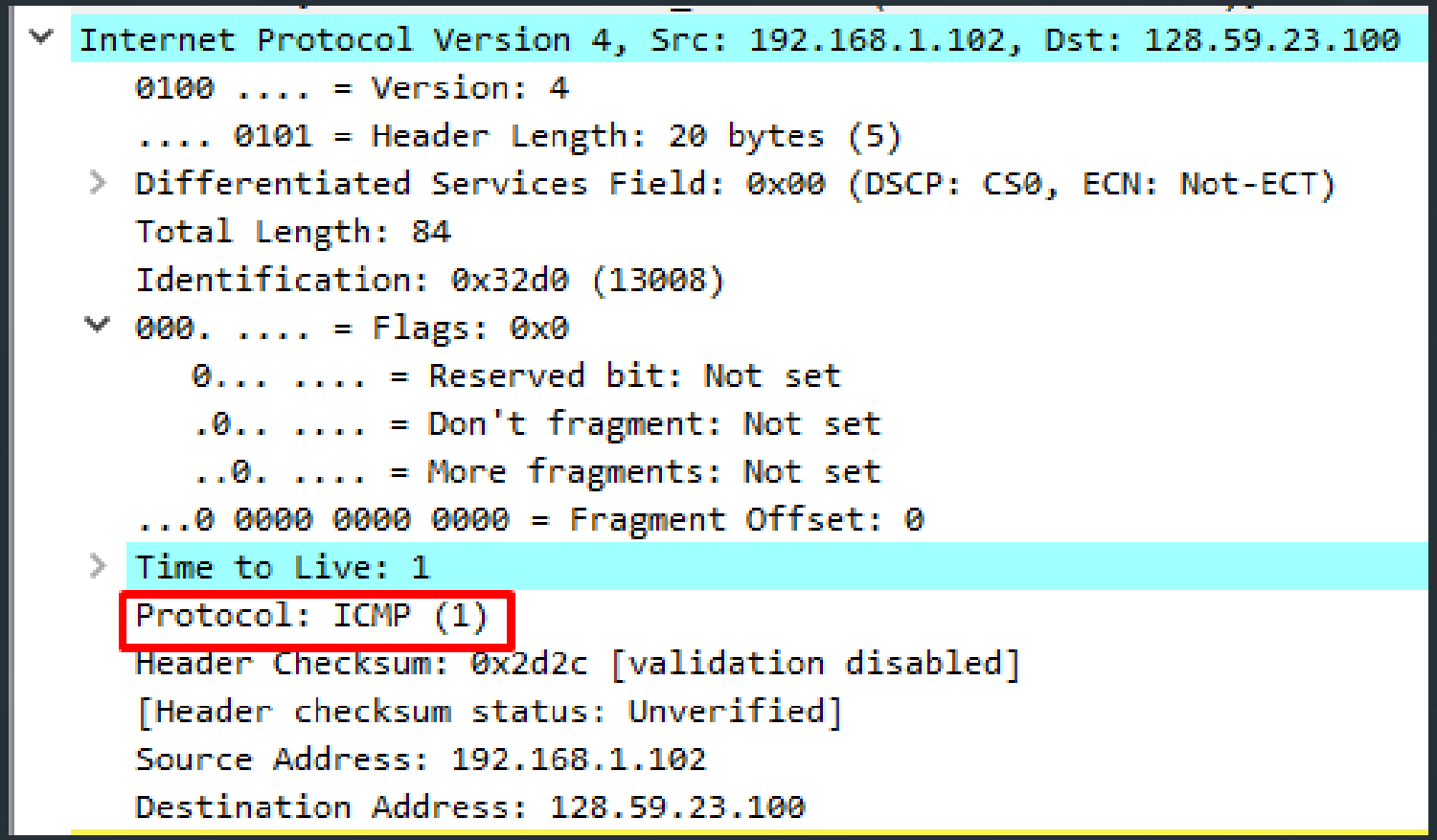
- Επιλέγουμε το πρώτο ICMP Echo Request μήνυμα



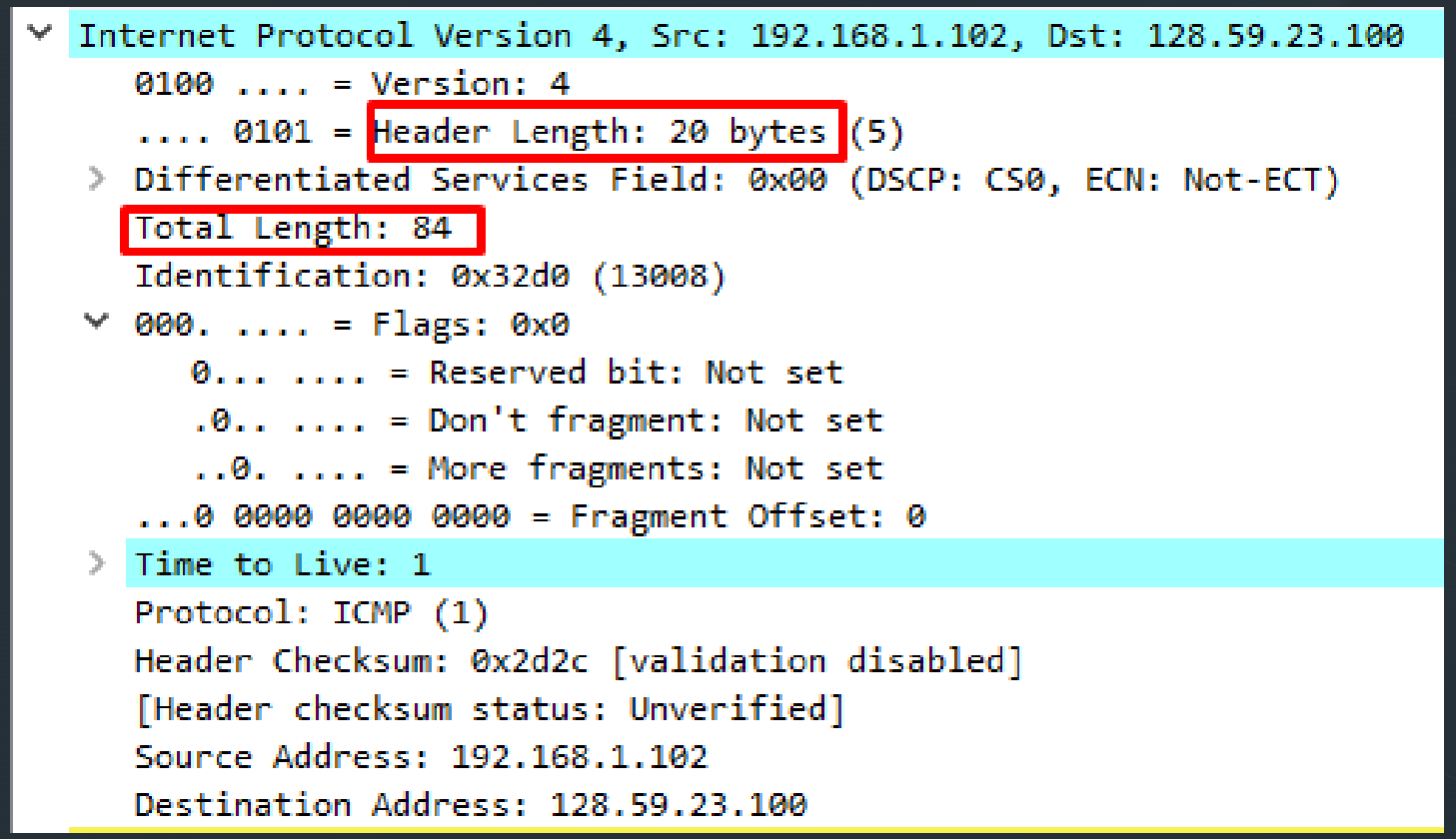
ποια είναι η IP διεύθυνση του υπολογιστή σας;



μέσα στην επικεφαλίδα, ποιο ανώτερο πρωτόκολλο περιέχεται;

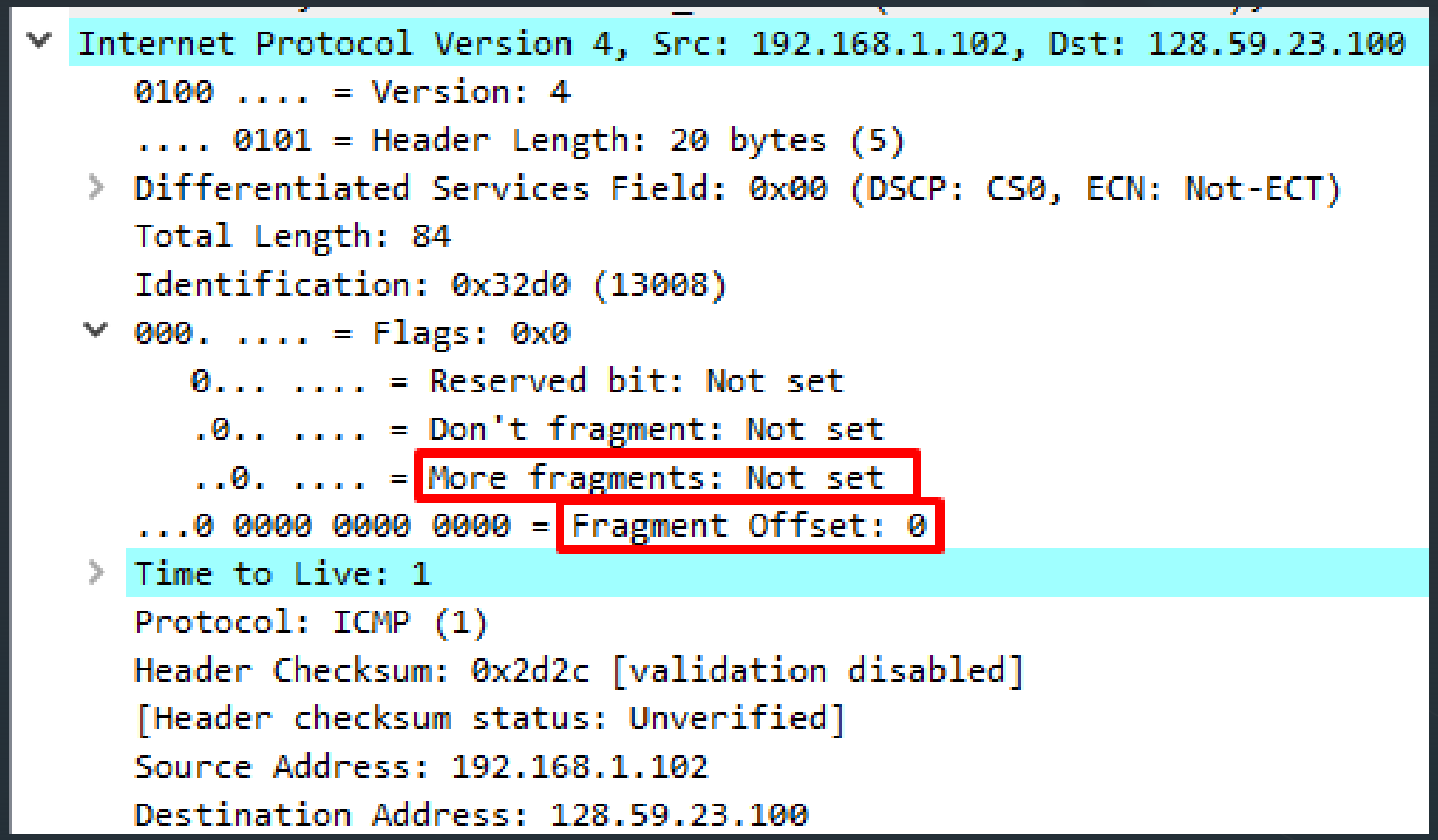


ποιο το μέγεθος (σε bytes) της IP επικεφαλίδας; ποιο το μέγεθος του IP payload;



payload: 64

έχει γίνει fragmentation; πως το καταλαβαίνουμε;



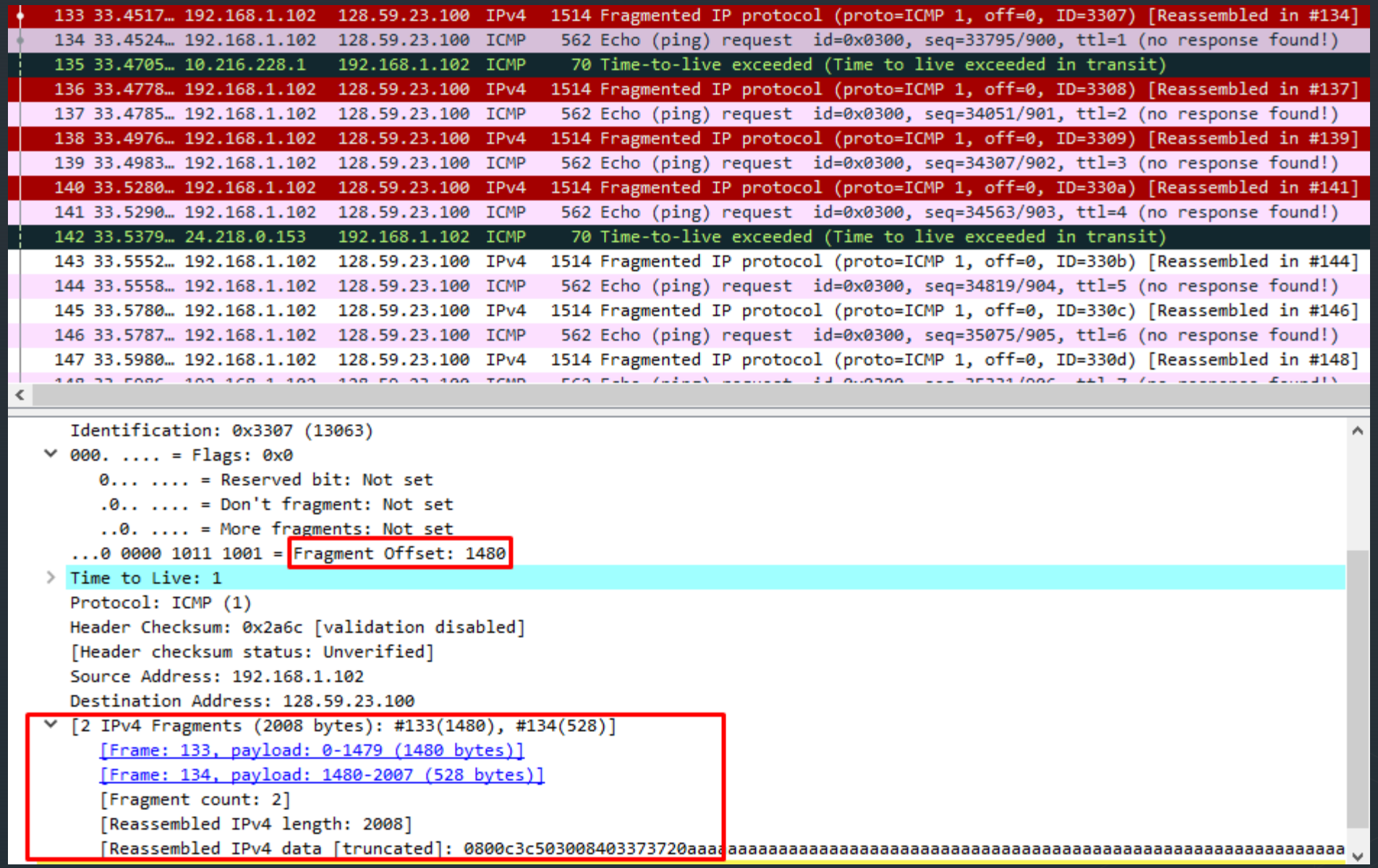
Παρατηρήστε την αλληλουχία των ICMP πακέτων που έχουν σταλεί από τον υπολογιστή σας

- ποια πεδία του IP datagram αλλάζουν από το ένα datagram στο άλλο;

traceroute με μέγεθος πακέτου 2000 bytes (για να αναγκάσουμε σε fragmentation)

132	32.0670…	192.168.1.102	199.2.53.206	TCP	62	1483 → 631 [SYN] Seq=0 Win=16384 Len=0 MSS=1460 SACK_PERM
133	33.4517…	192.168.1.102	128.59.23.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3307) [Reassembled in #134]
134	33.4524…	192.168.1.102	128.59.23.100	ICMP	562	Echo (ping) request id=0x0300, seq=33795/900, ttl=1 (no response found!)
135	33.4705…	10.216.228.1	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
136	33.4778…	192.168.1.102	128.59.23.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3308) [Reassembled in #137]
137	33.4785…	192.168.1.102	128.59.23.100	ICMP	562	Echo (ping) request id=0x0300, seq=34051/901, ttl=2 (no response found!)
138	33.4976…	192.168.1.102	128.59.23.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=3309) [Reassembled in #139]
139	33.4983…	192.168.1.102	128.59.23.100	ICMP	562	Echo (ping) request id=0x0300, seq=34307/902, ttl=3 (no response found!)
140	33.5280…	192.168.1.102	128.59.23.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=330a) [Reassembled in #141]
141	33.5290…	192.168.1.102	128.59.23.100	ICMP	562	Echo (ping) request id=0x0300, seq=34563/903, ttl=4 (no response found!)
142	33.5379…	24.218.0.153	192.168.1.102	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
143	33.5552…	192.168.1.102	128.59.23.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=330b) [Reassembled in #144]
144	33.5558…	192.168.1.102	128.59.23.100	ICMP	562	Echo (ping) request id=0x0300, seq=34819/904, ttl=5 (no response found!)
145	33.5780…	192.168.1.102	128.59.23.100	IPv4	1514	Fragmented IP protocol (proto=ICMP 1, off=0, ID=330c) [Reassembled in #146]
146	33.5787…	192.168.1.102	128.59.23.100	ICMP	562	Echo (ping) request id=0x0300, seq=35075/905, ttl=6 (no response found!)

```
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1500
    Identification: 0x3307 (13063)
  v 001. .... = Flags: 0x1, More fragments
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..1. .... = More fragments: Set
    ...0 0000 0000 0000 = Fragment Offset: 0
  > Time to Live: 1
    Protocol: ICMP (1)
    Header Checksum: 0x076d [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.102
    Destination Address: 128.59.23.100
    [Reassembled IPv4 in frame: 134]
```

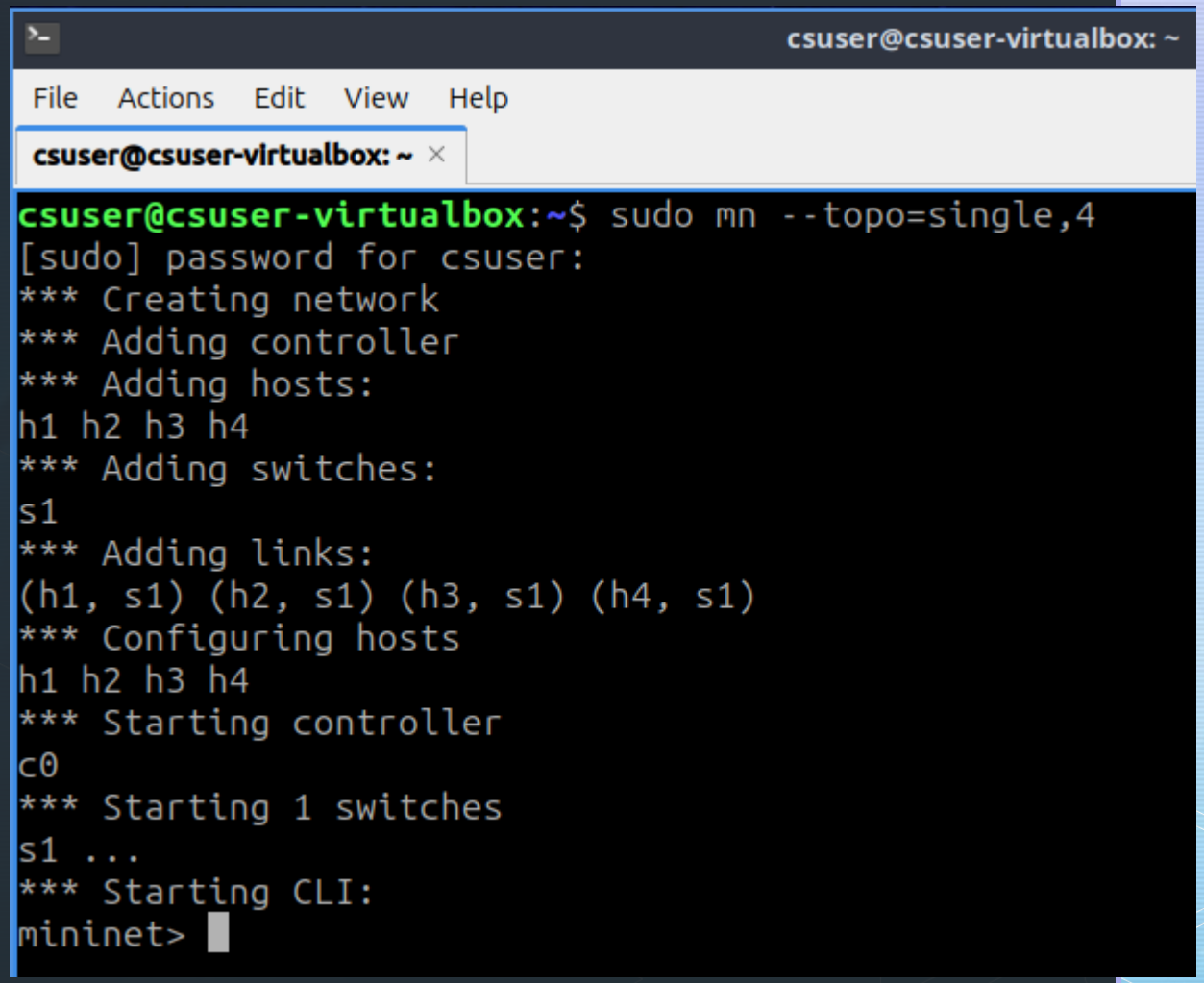

133 33.4517… 192.168.1.102 128.59.23.100 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=3307) [Reassembled in #134]
134 33.4524… 192.168.1.102 128.59.23.100 ICMP 562 Echo (ping) request id=0x0300, seq=33795/900, ttl=1 (no response found!)
135 33.4705… 10.216.228.1 192.168.1.102 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
136 33.4778… 192.168.1.102 128.59.23.100 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=3308) [Reassembled in #137]
137 33.4785… 192.168.1.102 128.59.23.100 ICMP 562 Echo (ping) request id=0x0300, seq=34051/901, ttl=2 (no response found!)
138 33.4976… 192.168.1.102 128.59.23.100 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=3309) [Reassembled in #139]
139 33.4983… 192.168.1.102 128.59.23.100 ICMP 562 Echo (ping) request id=0x0300, seq=34307/902, ttl=3 (no response found!)
140 33.5280… 192.168.1.102 128.59.23.100 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=330a) [Reassembled in #141]
141 33.5290… 192.168.1.102 128.59.23.100 ICMP 562 Echo (ping) request id=0x0300, seq=34563/903, ttl=4 (no response found!)
142 33.5379… 24.218.0.153 192.168.1.102 ICMP 70 Time-to-live exceeded (Time to live exceeded in transit)
143 33.5552… 192.168.1.102 128.59.23.100 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=330b) [Reassembled in #144]
144 33.5558… 192.168.1.102 128.59.23.100 ICMP 562 Echo (ping) request id=0x0300, seq=34819/904, ttl=5 (no response found!)
145 33.5780… 192.168.1.102 128.59.23.100 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=330c) [Reassembled in #146]
146 33.5787… 192.168.1.102 128.59.23.100 ICMP 562 Echo (ping) request id=0x0300, seq=35075/905, ttl=6 (no response found!)
147 33.5980… 192.168.1.102 128.59.23.100 IPv4 1514 Fragmented IP protocol (proto=ICMP 1, off=0, ID=330d) [Reassembled in #148]
Identification: 0x3307 (13063)
000. .... = Flags: 0x0
0... .... = Reserved bit: Not set
.0.. .... = Don't fragment: Not set
..0. .... = More fragments: Not set
...0 0000 1011 1001 = Fragment Offset: 1480
Time to Live: 1
Protocol: ICMP (1)
Header Checksum: 0x2a6c [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.1.102
Destination Address: 128.59.23.100
[2 IPv4 Fragments (2008 bytes): #133(1480), #134(528)]
[Frame: 133, payload: 0-1479 (1480 bytes)]
[Frame: 134, payload: 1480-2007 (528 bytes)]
[Fragment count: 2]
[Reassembled IPv4 length: 2008]
[Reassembled IPv4 data [truncated]: 0800c3c503008403373720aaa

WireShark

Mininet Demo

Έναρξη του Mininet

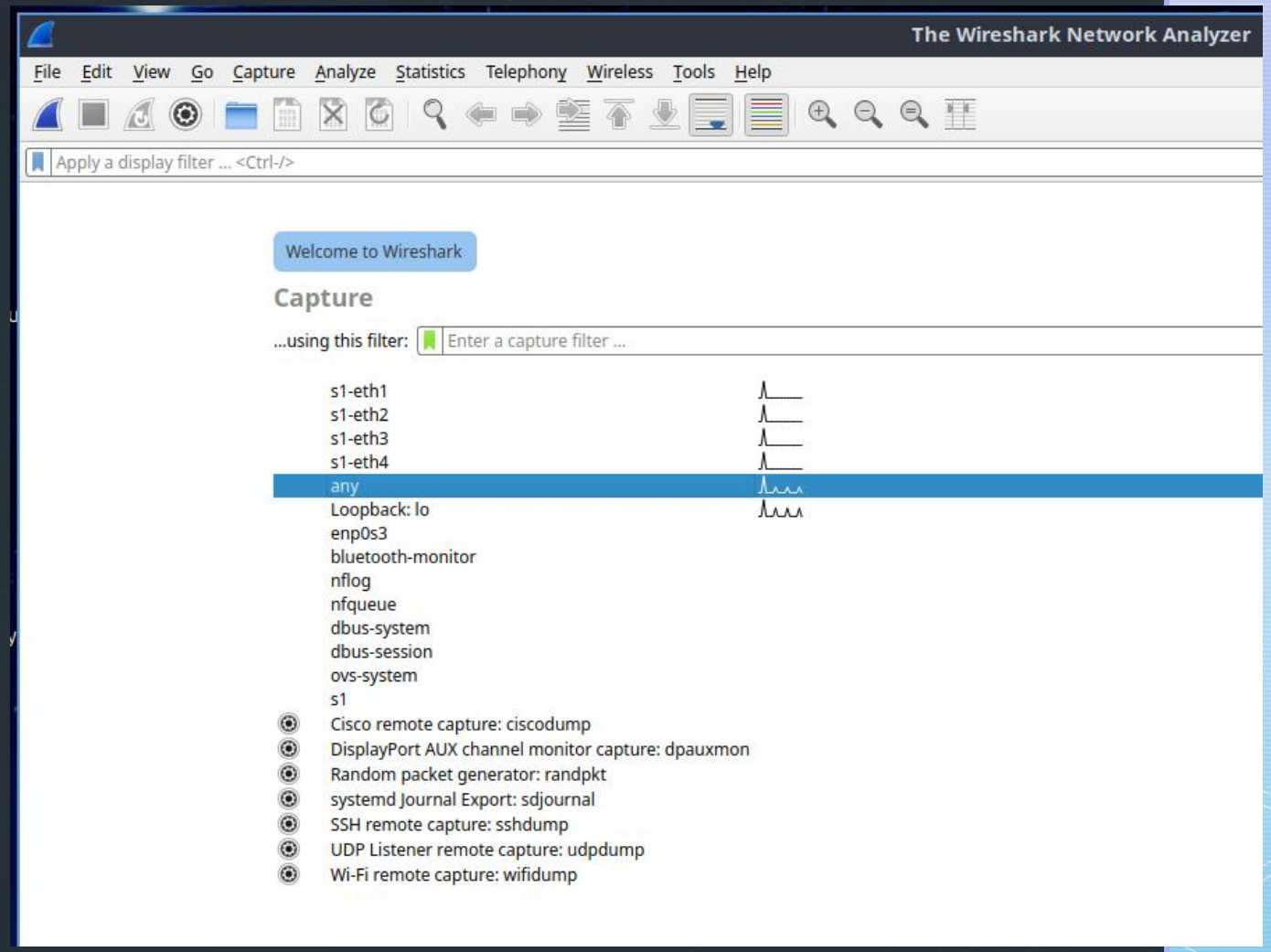
- sudo mn --topo=single,4



Έναρξη του WireShark (χωριστό terminal)

- sudo wireshark

- Επιλέγω interface: any
- Ξεκινάω capture



mininet> pingall

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/1
mininet>
```

*any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

No.	Time	Source	Destination	Protocol	Length	Info
23	12.719049434	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0x09cc, seq=1/256, ttl=64 (no response found!)
28	12.719865469	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) request id=0x09cc, seq=1/256, ttl=64 (reply in 29)
29	12.719891242	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0x09cc, seq=1/256, ttl=64 (request in 28)
34	12.720967910	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) reply id=0x09cc, seq=1/256, ttl=64
47	12.734443540	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) request id=0x09cd, seq=1/256, ttl=64 (no response found!)
52	12.735409060	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) request id=0x09cd, seq=1/256, ttl=64 (reply in 53)
53	12.735430606	10.0.0.3	10.0.0.1	ICMP	100	Echo (ping) reply id=0x09cd, seq=1/256, ttl=64 (request in 52)
58	12.735949721	10.0.0.3	10.0.0.1	ICMP	100	Echo (ping) reply id=0x09cd, seq=1/256, ttl=64
71	12.740819209	10.0.0.1	10.0.0.4	ICMP	100	Echo (ping) request id=0x09ce, seq=1/256, ttl=64 (no response found!)
76	12.741501491	10.0.0.1	10.0.0.4	ICMP	100	Echo (ping) request id=0x09ce, seq=1/256, ttl=64 (reply in 77)
77	12.741523712	10.0.0.4	10.0.0.1	ICMP	100	Echo (ping) reply id=0x09ce, seq=1/256, ttl=64 (request in 76)
82	12.742067482	10.0.0.4	10.0.0.1	ICMP	100	Echo (ping) reply id=0x09ce, seq=1/256, ttl=64
83	12.746391618	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) request id=0x09cf, seq=1/256, ttl=64 (no response found!)
88	12.746740180	10.0.0.2	10.0.0.1	ICMP	100	Echo (ping) request id=0x09cf, seq=1/256, ttl=64 (reply in 89)
89	12.746755820	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) reply id=0x09cf, seq=1/256, ttl=64 (request in 88)
94	12.747377160	10.0.0.1	10.0.0.2	ICMP	100	Echo (ping) reply id=0x09cf, seq=1/256, ttl=64
107	12.751447773	10.0.0.2	10.0.0.3	ICMP	100	Echo (ping) request id=0x09d0, seq=1/256, ttl=64 (no response found!)
112	12.752052007	10.0.0.2	10.0.0.3	ICMP	100	Echo (ping) request id=0x09d0, seq=1/256, ttl=64 (reply in 113)
113	12.752068606	10.0.0.3	10.0.0.2	ICMP	100	Echo (ping) reply id=0x09d0, seq=1/256, ttl=64 (request in 112)
118	12.753363006	10.0.0.3	10.0.0.2	ICMP	100	Echo (ping) reply id=0x09d0, seq=1/256, ttl=64
131	12.757944176	10.0.0.2	10.0.0.4	ICMP	100	Echo (ping) request id=0x09d1, seq=1/256, ttl=64 (no response found!)
136	12.758522106	10.0.0.2	10.0.0.4	ICMP	100	Echo (ping) request id=0x09d1, seq=1/256, ttl=64 (reply in 137)
137	12.758544257	10.0.0.4	10.0.0.2	ICMP	100	Echo (ping) reply id=0x09d1, seq=1/256, ttl=64 (request in 136)
142	12.758906728	10.0.0.4	10.0.0.2	ICMP	100	Echo (ping) reply id=0x09d1, seq=1/256, ttl=64
143	12.762671038	10.0.0.3	10.0.0.1	ICMP	100	Echo (ping) request id=0x09d2, seq=1/256, ttl=64 (no response found!)
148	12.763412673	10.0.0.3	10.0.0.1	ICMP	100	Echo (ping) request id=0x09d2, seq=1/256, ttl=64 (reply in 149)
149	12.763428793	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) reply id=0x09d2, seq=1/256, ttl=64 (request in 148)
154	12.764376733	10.0.0.1	10.0.0.3	ICMP	100	Echo (ping) reply id=0x09d2, seq=1/256, ttl=64
155	12.767641238	10.0.0.3	10.0.0.2	ICMP	100	Echo (ping) request id=0x09d3, seq=1/256, ttl=64 (no response found!)
160	12.768067965	10.0.0.3	10.0.0.2	ICMP	100	Echo (ping) request id=0x09d3, seq=1/256, ttl=64 (reply in 161)
161	12.768085566	10.0.0.2	10.0.0.3	ICMP	100	Echo (ping) reply id=0x09d3, seq=1/256, ttl=64 (request in 160)
166	12.768674430	10.0.0.2	10.0.0.3	ICMP	100	Echo (ping) reply id=0x09d3, seq=1/256, ttl=64

```
▸ Frame 23: 100 bytes on wire (800 bits), 100 bytes captured (800 bits)
▸ Linux cooked capture v1
▸ Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
▸ Internet Control Message Protocol
```

```
0000  00 03 00 01 00 06 26 40  3a a5 bb 40 00 00 08 0
0010  45 00 00 54 fe 47 40 00  40 01 28 5f 0a 00 00 0
0020  0a 00 00 02 08 00 2d 74  09 cc 00 01 bd 99 62 6
0030  00 00 00 00 d9 ea 08 00  00 00 00 00 10 11 12 1
0040  14 15 16 17 18 19 1a 1b  1c 1d 1e 1f 20 21 22 2
0050  24 25 26 27 28 29 2a 2b  2c 2d 2e 2f 30 31 32 3
0060  34 35 36 37
```